

# Internet infrastructure

Prof. dr. ir. André Mariën

# Domain Naming System

# Introducing names

- People prefer names instead of numbers
- Initial system: per host
  - “hosts” on Unix
  - “hosts” on Windows
- Information: table with IP – host mapping

# Network information

- What information is needed?
  - host - IP mapping
  - router information
  - subnet information
  - ... See dynamic host configuration
- Problem: copy per host: change management
- Solution: network information service
  - First: yellow pages (yp\*\*\* programs)
  - Then: Network Information System (NIS)

# Domain Naming System

- DNS: domain naming system
- Managers of information
  - registration: name to IP mapping
  - name service servers (example: bind)
- Consumers of information
  - name lookup service: name to IP
  - reverse look-up: IP to name

# Definition of DNS

- RFC 1034: STD 13: Domain names - Concepts and Facilities  
note: November 1987!
- RFC 1035: STD 13: Domain Names - Implementation and Specification
- RFC 2065: Domain Name System Security Extensions
- RFC 2181: Clarifications to the DNS Specification

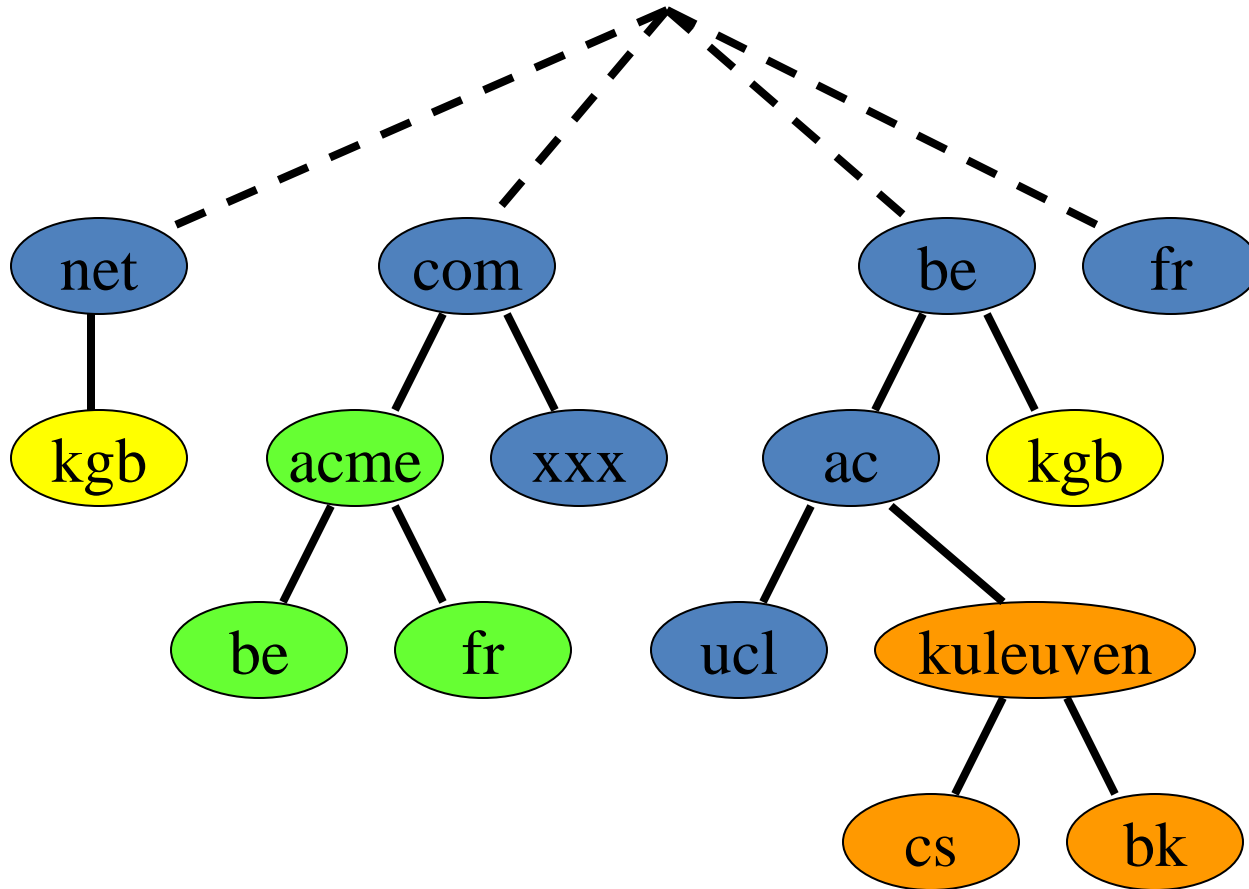
# Naming - Networking

- Names are independent from network operation
- DNS is a service on top of IP (TCP/UDP)
- Totally different naming system possible without network impact
- Applications prefer naming interface
  - Ex: URLConnection vs. socket

# Naming system

- Forest structure
- Limited number of trees
  - US names: .gov, .mil, .edu, .org, .com, .net, .int
  - ISO country code names: .be, .ca
- Distributed responsibility
- Each top level domain has its own structure
  - United Kingdom: .co.uk, .ac.uk
  - Belgium: .ac.be, but no .co.be

# Forest



# Naming system

- .acme.com:
  - subdivided: .be.acme.com, fr.acm.com
  - compare: acme.be, acme.fr
- Trade-off:
  - structure: clear ordering
  - short names: easy to remember and find
    - (kuleuven.ac.be to kuleuven.be)

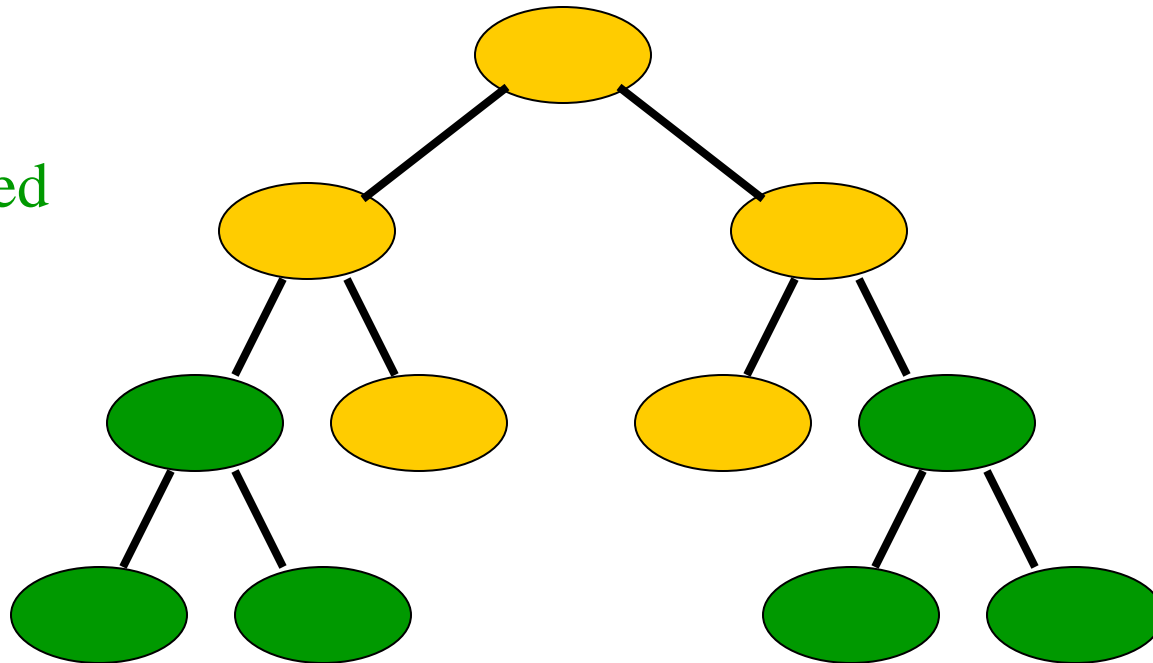
# Naming management

- Registration:
  - owning a domain costs money
  - regulations: local to domain
- Delegation:
  - subdomain responsibility: someone else
- Zone
  - naming information within scope of one name server

# Zone

Zone

Delegated



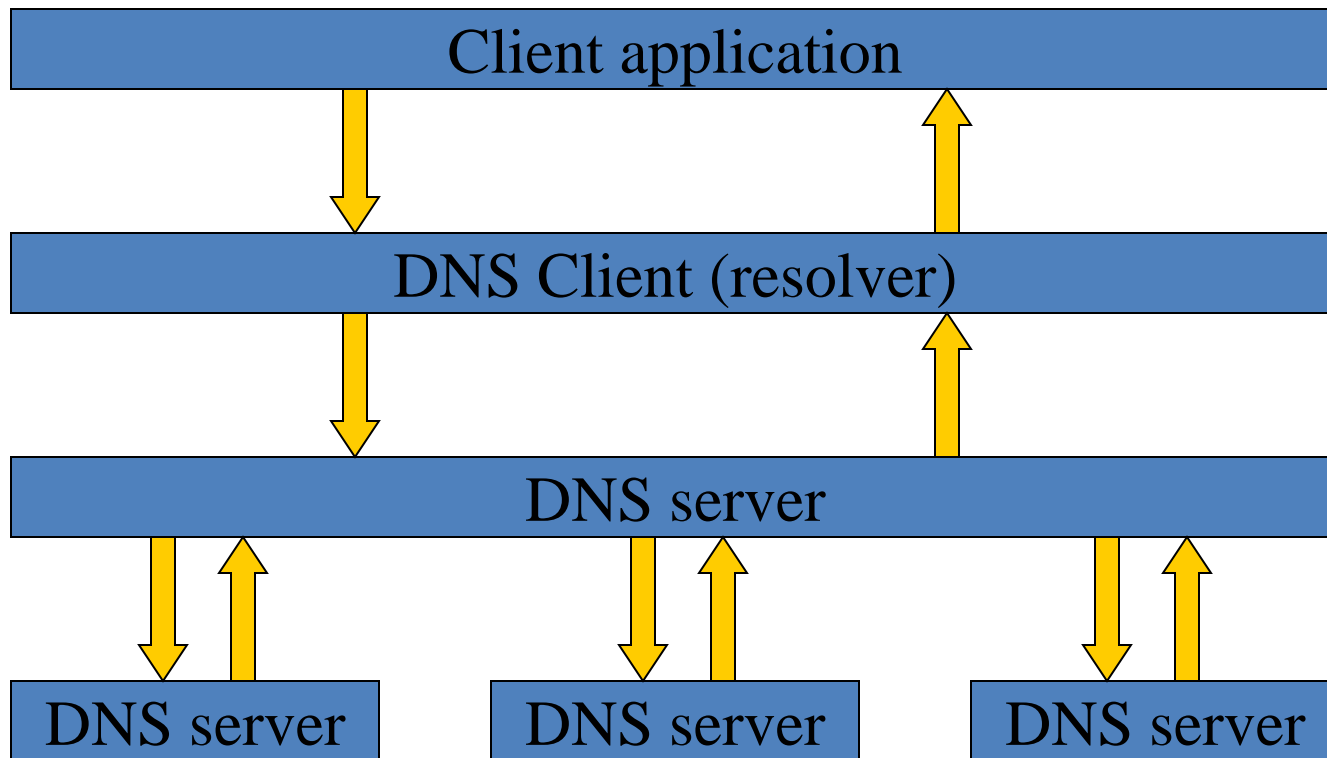
# DNS: distributed client - server

- Simple system:
  - clients: request name-to-IP translation
  - server: looks up mapping, returns all answers
- More complex system:
  - server is responsible for zone
  - if request cannot be handled, look for answer on other servers (*recursive*)

# Other type of requests

- IP to name mapping
  - Reverse DNS look-up
  - Verification: connection from right place
    - Weak protection
    - .com, .org, .net: no geographic information
- Mail exchange info
  - MX records (TBD in the e-mail part)
- Authority information (SOA)

# DNS server cooperation



# Root name servers

- At least one root server per top level domain
- In principle: need to start from there to find anything
- IP addresses of those servers should be stable
- Locations of the root servers must be configured in name servers
- Replies can be (should be) cached

# Reverse DNS 'hack'

- Reverse DNS: complementary forest structure
- Made to look a lot like the name-to-IP structure
- Naming root: in-addr.arpa
- Next level: highest order IP address byte
- Example:
  - IP: 163.7.23.89
  - “reverse name”: 89.23.7.163.in-addr.arpa

# DNS performance

- Critical internet infrastructure:
  - Each name-based request needs resolving
- Any server in the world needs to be mapped quickly from anywhere
- Solutions
  - Caching: local, organization, ISP, ...
  - Quick homing into “right” server via referral

# Get running: domain name and ISP

- Need an ISP to connect
- ISP rents range of IP addresses
- Need to decide on parent domain
- Need to select top domain name (regulations)
- Need to decide to run own server or use ISP's
- Need to register top domain name

# Set up primary DNS server

- Define server parameters (time-outs)
- Define name to IP mapping
- Define IP to name mapping
- Configure top level server locations

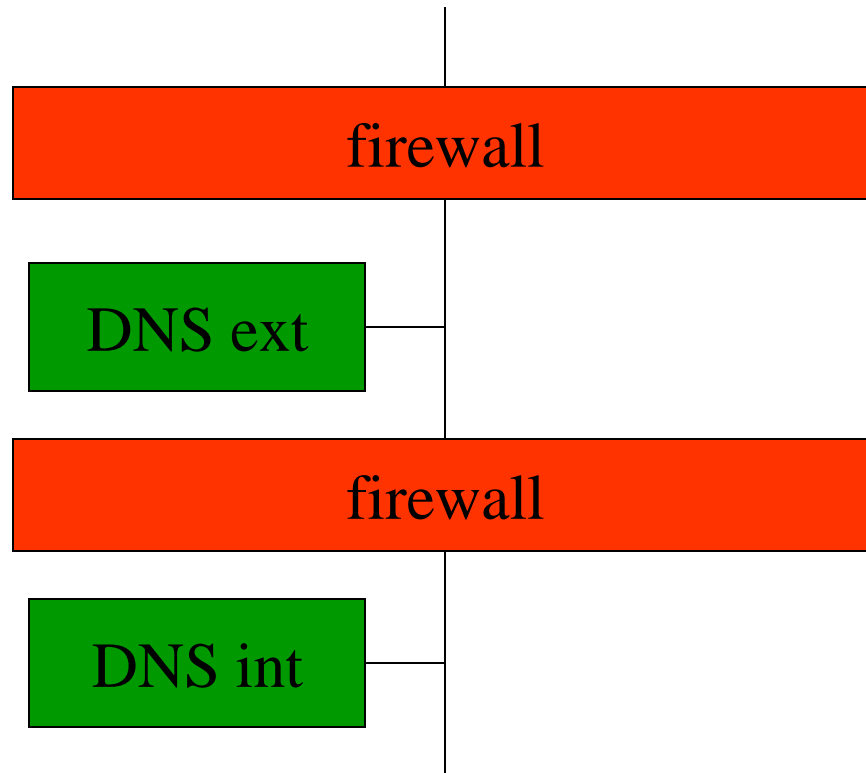
# Secondary DNS server

- Typically at different network location
- Copies data from primary DNS server (=zone transfer)
- Synchronization: uses SOA information
  - modification: SERIAL number
  - polling: REFRESH
  - RETRY: connectivity problems
  - time-out: EXPIRE

# Internal – external DNS

- DNS can be used by hackers to investigate remote systems
  - Zone transfers: all servers
  - Reverse DNS mapping (range of IP addresses)
- Risk is high if internal/external mappings are handled by one server
- Advice: split DNS in internal/external

# Example set-up



# Information in DNS: Resource Records (RRs)

- name to address
- address to name
- nick names
- host information (security)
- DNS servers (delegation)
- Mail eXchange

# DNS carrier

- protocol: TCP or UDP
  - UDP: typical name lookup queries
  - TCP:
    - zone transfer
    - queries with long replies
- TCP/53 or UDP/53

# Tools for DNS querying

- whois
- nslookup
- nstest: diagnostic tool
- Dig

# Information sources

- **RIPE (Réseaux IP Européens):**
  - <http://www.ripe.net/db/index.html>
  - <http://www.db.ripe.net/whois>
- Arin: american registry for internet numbers
  - `whois -h rr.arin.net <object>`
- Asia Pacific network information center
  - <http://wq.apnic.net/apnic-bin/whois.pl>
- African Internet Community
  - <http://www.afrinic.net/cgi-bin/whois>

# nslookup

- basic mode
  - *nslookup name*
  - uses your configured DNS server
- telling which DNS server to use
  - *nslookup name dnsserver*
  - recursive queries may not be allowed on other DNS servers than “your” server
- Note: DNS servers may not allow recursive queries for everyone, just zone enquiries

# nslookup interactive

- default: recursive queries (*[no]recurse*)
- default server (*server <dnsserver>*)
- querytypes: default ANY
- zone transfer
  - *ls <domain>*
  - Note: often restricted

# Name server selection

- Query:
  - NAME  
print info about the host/domain NAME using default server
  - NAME1 NAMESERVER  
search NAME1, but use NAMESERVER as server
  - server NAME  
set default server to NAME, using current default server
  - lserver NAME  
set default server to NAME, using initial server
  - root  
set current default server to the root

# Options (set ...)

- [no]debug, [no]d2
  - [exhaustive] debugging info
- [no]defname
  - append domain name to query
- [no]recurse
  - recursive answer
- [no]vc
  - always use a virtual circuit

# Options (set ...)

- domain=NAME
  - set default domain
- srchlist=N1[/N2/.../N6]
  - set domain to N1 and search list to N1,N2, etc.
- root=NAME
  - set root server to NAME
- Flags
  - retry=X, timeout=X

# Set Query Type

- Set [query]type=<choose>
  - ANY,
  - A(ddress),P(oin)T(e)R,M(ail e)X(change),S(tart)O(f)A(uthority),N(ame)S(erver)
  - C(anonical)NAME,H(ost)INFO(rmation)
  - PX,TXT,WKS,SRV,NAPTR

# Domain listing (in theory)

- `ls [opt] DOMAIN [> FILE]`  
list addresses in DOMAIN (optional: output to FILE)
  - a - list canonical names and aliases
  - h - list HINFO (CPU type and operating system)
  - s - list well-known services
  - d - list all records
  - t TYPE - list records of the given type (e.g., A,CNAME,MX, etc.)
- `view FILE`  
sort an 'ls' output file and view it with “more”

# Example nslookup queries: tree descend

- >Com.
- <nameserver = E.GTLD-SERVERS.NET
- >google.com
- <nameserver=NS3.google.com
- >www.google.com
- >www.google.com
- <216.239.37.100

# Example nslookup queries: multiple addresses

Q> www.microsoft.com

**Server: dns.xxx.com**

**Address: 10.0.0.7**

Non-authoritative answer:

Old: Name: [www.microsoft.akadns.net](http://www.microsoft.akadns.net)

New: Name: lb1.www.ms.akadns.net

Old: Addresses: 207.46.197.100, 207.46.197.102, 207.46.230.218,  
207.46.197.113, 207.46.197.101, 207.46.230.219,  
207.46.230.220

New: Addresses: 207.46.193.254, 207.46.192.254,  
65.55.12.249

Old: Aliases: www.microsoft.com

New: www.microsoft.com, toggle.www.ms.akadns.net,  
g.www.ms.akadns.net

# example nslookup queries: name servers

Q> set querytype=any

[www.sun.com](http://www.sun.com)

internet address = 192.18.97.241

sun.com nameserver = ns.sun.com

...

sun.com nameserver = ns1.pr.sun.com

ns.sun.com internet address = 192.9.9.3

...

ns1.pr.sun.com internet address = 192.18.16.2

# Example nslookup queries: mail records

```
Q> set querytype=mx
```

```
Q> sun.com
```

```
sun.com preference = 40, mail exchanger = mx6.sun.com
```

```
sun.com preference = 5, mail exchanger = mx8.sun.com
```

```
sun.com nameserver = ns1.eu.sun.com
```

```
mx6.sun.com internet address = 192.9.22.1
```

```
mx8.sun.com internet address = 192.18.98.36
```

```
ns1.eu.sun.com internet address = 192.18.240.8
```

# Example nslookup queries: initial query

Q> set querytype=any

Q> www.ibm.com

www.ibm.com internet address = 129.42.16.99

www.ibm.com internet address = 129.42.17.99

www.ibm.com preference = 10, mail exchanger = mail.www.ibm.com

ibm.com nameserver = ns.watson.ibm.com

ibm.com nameserver = internet-server.zurich.ibm.com

mail.www.ibm.com internet address = 198.133.21.65

ns.watson.ibm.com internet address = 198.81.209.2

internet-server.zurich.ibm.com internet address = 195.212.119.252

# Example nslookup queries: query repeated

```
Q> set querytype=a
```

```
Q> www.ibm.com
```

```
Non-authoritative answer:
```

```
Name: www.ibm.com
```

```
Addresses: 129.42.18.99, 129.42.19.99, 129.42.16.99, 129.42.17.99
```

```
Q> www.sun.com
```

```
Non-authoritative answer:
```

```
Name: www.sun.com
```

```
Address: 192.18.97.241
```

# Example nslookup queries: reverse look-up

Q> set querytype=ptr

Q> 192.151.52.217

217.52.151.192.in-addr.arpa name = hpat949.external.hp.com

52.151.192.in-addr.arpa nameserver = atlrel1.hp.com

52.151.192.in-addr.arpa nameserver = palrel1.hp.com

atlrel1.hp.com internet address = 156.153.255.210

atlrel1.hp.com internet address = 15.10.176.10

palrel1.hp.com internet address = 156.153.255.242

palrel1.hp.com internet address = 15.81.168.10

# Example nslookup queries: aliases

```
Q> set querytype=any
```

```
Q> www.oracle.com
```

```
www.oracle.com canonical name = bigip-  
www.us.oracle.com
```

```
bigip-www.us.oracle.com internet address = 148.87.9.44
```

```
oracle.com nameserver = ns1.oracle.com
```

```
oracle.com nameserver = udns1.ultradns.net
```

```
ns1.oracle.com internet address = 148.87.1.20
```

# Example nslookup queries: ask authoritative server

Q> www.oracle.com ns1.oracle.com

Server: ns1.oracle.com

Address: 148.87.1.20

www.oracle.com canonical name = bigip-  
www.us.oracle.com

oracle.com nameserver = ns1.oracle.com

oracle.com nameserver = udns1.ultradns.net

ns1.oracle.com internet address = 148.87.1.20

udns1.ultradns.net internet address = 204.69.234.1

# Example nslookup queries: cached answer

Q> www.oracle.com

Non-authoritative answer:

www.oracle.com canonical name = bigip-  
www.us.oracle.com

# Example nslookup queries: 9 (cont.)

Authoritative answers can be found from:

oracle.com    nameserver = ns1.oracle.com

oracle.com    nameserver = udns1.ultradns.net

ns1.oracle.com internet address = 148.87.1.20

udns1.ultradns.net    internet address = 204.69.234.1

# Example nslookup queries: aliases

```
Q>set type=cname
```

```
Q> www.oracle.com ns1.oracle.com
```

```
Server: ns1.oracle.com
```

```
Address: 148.87.1.20
```

```
www.oracle.com canonical name =
```

```
bigip-www.us.oracle.com
```

```
oracle.com nameserver = ns1.oracle.com
```

```
ns1.oracle.com internet address = 148.87.1.20
```

# Example nslookup queries: mail

- Q> set querytype=mx
- Q> ac.be.
- Server: dns.xxx.com
- ac.be preference = 0, mail exchanger = mail.belnet.be
- ac.be nameserver = ns.belnet.be
- ac.be nameserver = ns.dns.be
- ac.be nameserver = ns1.surfnet.nl

# Example nslookup queries: IPv6

- mail.belnet.be IPv6 address =  
3ffe:80b0:0:1:a00:20ff:fea2:8dbc
- mail.belnet.be IPv6 address =  
2001:6a8:0:1:a00:20ff:fea2:8dbc
- mail.belnet.be internet address =  
193.190.198.2

# Example nslookup queries: IPv6

- ns.belnet.be IPv6 address =  
3ffe:80b0:0:1:a00:20ff:fea2:8dbc
- ns.belnet.be IPv6 address =  
2001:6a8:0:1:a00:20ff:fea2:8dbc
- ns.belnet.be internet address = 193.190.198.10
- ns.belnet.be internet address = 193.190.198.2
- ns.dns.be internet address = 134.58.74.33
- ns1.surfnet.nl internet address = 192.87.106.101

# Example nslookup queries: mail

- > kuleuven.ac.be.
- kuleuven.ac.be preference = 10, mail exchanger = krimson.cc.kuleuven.ac.be
- kuleuven.ac.be preference = 20, mail exchanger = lambik.cc.kuleuven.ac.be
- kuleuven.ac.be preference = 30, mail exchanger = urc1.cc.kuleuven.ac.be

# Example nslookup queries: (cont.)

- kuleuven.ac.be nameserver = ns1.kulnet.kuleuven.ac.be
- kuleuven.ac.be nameserver = ns2.kulnet.kuleuven.ac.be
- kuleuven.ac.be nameserver = ns.be.ubizen.com
- kuleuven.ac.be nameserver = ns2.sri.ucl.ac.be

# Example nslookup queries: (Cont.)

- `krimson.cc.kuleuven.ac.be` internet address = `134.58.10.5`
- `lambik.cc.kuleuven.ac.be` internet address = `134.58.10.1`
- `urc1.cc.kuleuven.ac.be` internet address = `134.58.10.3`
- `ns1.kulnet.kuleuven.ac.be` internet address = `134.58.126.3`
- `ns2.kulnet.kuleuven.ac.be` internet address = `134.58.127.1`

# Example nslookup queries: select nameserver via IP

```
Q> 134.58.45.30 134.58.41.8
```

```
Server: [134.58.41.8]
```

```
30.45.58.134.in-addr.arpa name = idfix.cs.kuleuven.ac.be
```

```
45.58.134.in-addr.arpa nameserver = ns1.kulnet.kuleuven.ac.be
```

```
idfix.cs.kuleuven.ac.be internet address = 134.58.45.30
```

```
ns1.kulnet.kuleuven.ac.be internet address = 134.58.126.3
```

```
dns.cs.kuleuven.ac.be internet address = 134.58.40.4
```

# Example nslookup queries: Reverse mapping tree

Q> 134.in-addr.arpa.

primary name server = arrowroot.arin.net

134.in-addr.arpa nameserver = HENNA.arin.net

134.in-addr.arpa nameserver = INDIGO.arin.net

...

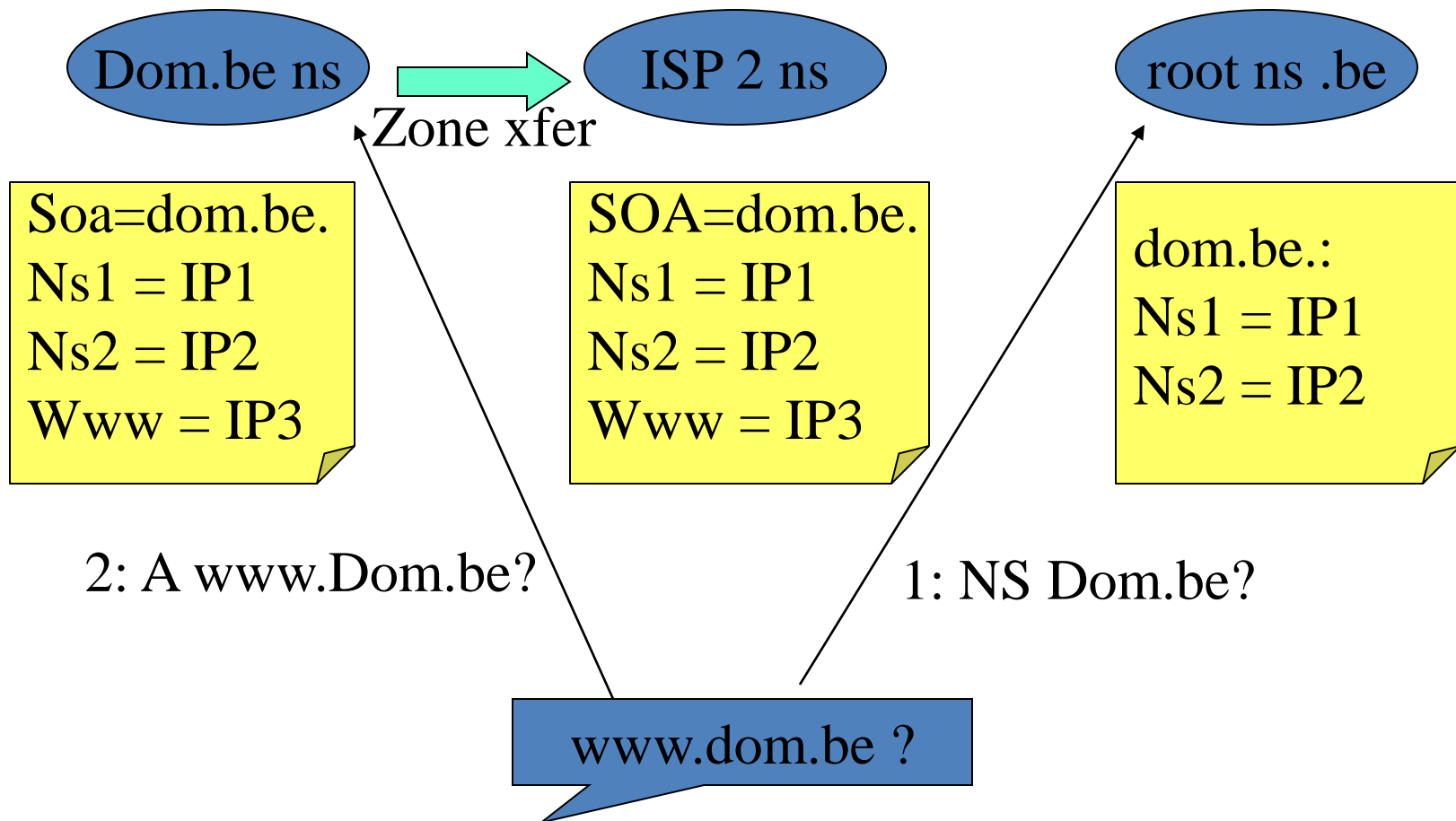
Q> 58.134.in-addr.arpa

primary name server = ns1.kulnet.kuleuven.ac.be

Q> 59.134.in-addr.arpa

primary name server = taloa.unice.fr

# DNS system



# Exercise

- Compare DNS and ARP protocols

# DNSSecure

# DNSecure

- Why DNSSecure? DNS is very insecure!
  - UDP based
  - no authentication
  - enables man-in-the-middle attacks
- Definition of DNSSecure?
  - RFC 2535: DNS Security extensions

# Security risk

- Denial of Service (DoS)
- Man in the middle (MITM)
- Domain intrusion
  - Authentication via IP, reverse DNS
  - Cookies set for a domain

# Which Security Measures?

- Authentication
  - data
  - request
  - transaction (request+reply)
- Integrity
  - indirect, via authentication system
- Not: confidentiality
- Not: authorization (ACL or other)

# Mechanism: signatures

- public key technology
- key distribution: via DNS
  - Two new RRs
    - KEY RR: signed public keys
    - SIG RR: signatures

# Signatures

- sign Resource Record sets + validation
- signer: zone key
- pre-signing: data authentication

# Trust

- trust hierarchy: zone signs subzone keys
- untrusted subzones: zone signs 'no key' KEY RR

# NOT FOUND authentication

- Mechanism:
  - chain of authenticated data
  - signed response: before - after RR indicates data not there
  - uses NXT RR
  - based on canonical ordering of names
  - end marker: first name: zone itself

# Multiple keys

- one key (pair) per technology
- difference between
  - zone keys: data authentication
  - host keys: transaction or request authentication

# KEY RR

- keys are labeled for use:
  - zone key: x.y : zone x.y
  - server key: www.x.y : server www in zone x.y
  - user key: a.x.y : user a@x.y
- key used in protocol: DNSSec, IPSec, ...
- keying algorithm: RSA/MD5, DH, DSA, ...

# DDNS

# Problem

- **DNS automates DNS database updates**
- Dynamic Host Configuration Protocol ([DHCP](#))
  - might assign a host only a temporary address, requiring many different addresses in succession
  - Most computers at a site receive IP addresses via DHCP
  - Result:
    - many and frequent IP address changes
    - manual DNS administration impractical.

# Solution

- extend DNS to accommodate dynamic networking environments
- Dynamic DNS (DDNS)
  - An umbrella term for three related DNS protocol extensions:
    - Dynamic Update
    - Notify (RFC 1996)
    - Incremental Zone Transfer (IXFR), (RFC 1995)

# Managing zone data

- Old style:
  - editing text files
  - add A (address) and PTR (pointer)
- Dynamic Update:
  - The basic DDNS operation
  - permits DHCP clients/servers to send special messages to name servers to update the data.

# Notify

- TTL of DNS data in DNS caches is a problem (updates are delayed)
- Solution: Notify message
  - the primary name server notifies the secondary name servers that the contents of a particular zone have changed.
  - NOTIFY request type, indicating the nature of what has changed
  - Uses the version number of the data
- NS records identify who to notify

# Incremental operation

- Problem: many notifications on large networks
- Solution: incremental zone transfer (IXFR)
  - DNS request, type=IXFR
  - Contains current SOA record for the zone (including version number)
  - Response includes deleted and new RRs
- DNS server reacts to notification
  - I have version x, please send differences only
  - Master server sends delta since that version, or all if too old
    - Must maintain a delta list

# References

- DNS and BIND, 4th Edition  
By Paul Albitz, Cricket Liu  
4th Edition April 2001  
0-596-00158-4  
622 pages
- DNS on Windows 2000  
By Matt Larson, Cricket Liu  
2nd Edition September 2001  
0-596-00230-0, 349 pages

# References

- <http://www.dns.net/dnsrd/rfc/>: DNS related RFCs
- <http://www.domtools.com/dns/>
- <http://www.sampade.org/ssw/features.html>